TI-59 FIRMWARE
by
Palmer O. Hanson, Jr.

Introduction

In early 1980 a personal letter from Maurice Swinnen, the editor of *TI PPC Notes* explained that the TI-59 firmware could be observed by performing the following unusual keyboard sequence while disregarding the flashing display:

Step	Display
Turnon	0
2nd-Op-09	Flashing 0.
2nd-Pgm-24	Flashing 0.
R/S	Flashing 0
R/S	Flashing 1
R/S	Flashing 0
9-2nd-Op-17	Flashing 239.89
GTO-0	Flashing 239.89
R/S	Flashing 0. 00
2nd-DMS	Flashing 0
LRN	000 82

One may then single step (SST) through the firmware up to location 575. Pressing almost any other key seems to return the calculator to the normal mode. After reaching location 575 an additional SST yields a zero in the display. A second additional SST yields an 04 in the display. A third additional SST yields a "locked-up" mode with the faint [on the far left. Exit from that condition can only be obtained by turning the calculator off.

Examination of the instructions obtained from the SST sequence will show that the code from locations 512 through 575 repeats the code in locations 384 through 447. Pages 16 through 19 show the listing through location 511. The listing shows rational program steps to mechanize the statistics and conversions functions in program locations 000 through 379. The apparent "garbage" instructions in the remaining locations have been shown to be the equivalent instructions to constants which would be useful in mechanizing the transcendental functions.

The original discovery of the ability to download the firmware was by Steffen Seitz (see Reference 1). Other calling sequences have been discovered which will also permit viewing of the firmware (see References 7 and 8). The additional calling sequences have not revealed any additional firmware, but have been helpful in understanding the calling techniques.

Analysis of the Firmware

Most of the statistics and conversions functions include manipulation of the hierarchy registers through the use of the HIR command (code 82). Volume 4 Number 3 of *PPX Exchange* dated May/June 1980 contains a discussion of most of the HIR options. The HIR-20 sequence which appears at locations 045/046 and 082/083 is not discussed in that reference. An analysis of the program sequences indicated that the HIR-20 command must be some kind of GTO-2nd-Ind instruction where the indirect address is somehow established by the entry to the firmware. For example, the HIR-20 command at locations 045/046 must result ins

A RTN when performing Op-12, or

A do-nothing and continue the calculations at location 047 for Op-15, or

A jump to location 058 for Op-14.

I wrote to Maurice Swinnen asking whether he had any further information on the HIR-20 command. His reply was "*The HIR-20 in the firmware of the TI-59 is a mystery to everyone. ... They showed me on the emulator how HIR-20 executed THERE as a RTN. But it does not do it in user memory. Other HIR-2X combinations, such as HIR-21 etc., do not work either in the emulator or in TI-59 user memory.*"

The article on "Hard-wired Functions" on page 7 of the March/April 1981 issue of *PPX Exchange* (Reference 9) stated "As you will see, this instruction can act as Nop, GTO, or INV-SBR depending upon a condition preset by the algorithm before the sequence is called. The only effect this sequence has in main memory is to kill a live entry state."

Page 14 of Patent 4,153,937 (Reference" 6) describes HIR-20 as a "conditional return program code". Table IV of the patent indicates that the second digit of an HIR-2X command is ignored.

Angle Conversions DMS and INV-DMS

Locations 303 through 340 of the firmware mechanize the DMS function. Locations 341 through 379 mechanize the INV-DMS function. One would have expected that the mechanizations would include an EE-INV-EE sequence at entry to limit the calculations to the displayed value only as noted on page V-30 of Personal Programming. In the absence of such a sequence it seems that there must be an implicit equivalent built into the DMS command.

Polar/Rectangular Conversions P→R and INV-P→R

Locations 284 through 302 mechanize the P→R function.
Locations 250 through 283 mechanize the INV-P→R function.

Statistics Data Entry Σ+ and INV-Σ+

The Σ+ function starts at location 192. The INV-Σ+ function starts at location 213. Both functions end at location 249.

Calculation of Mean Values (x_m)

The x_m function begins at location 067 and ends at the HIR-20 command at locations 082/083.

Calculation of Standard Deviations (INV x_m)

The INV x_m function begins at location 107 and ends at location 148. The equations are mechanized as indicated on page V-34 of *Personal Programming*.

Calculation of Variances (Op-11)

The Op 11 function begins at location 067 and uses the same instructions as those used for the x_m function to calculate the mean values; however, at the HIR-20 command at locations 082/083 the Op 11 function continues on to calculate the variances using the equations as indicated on page V-34 of *Personal Programming*. The Op 11 function ends at location 106.

Linear Regression

The Op 12 function begins at location 000 and ends at the HIR-20 command at locations 045/046. The solutions for the slope and intercept are mechanized as indicated on page V-37 of *Personal Programming*. The slope is calculated first, and the value of the slope is used in the calculation of the intercept.

The Op 13 function which computes the correlation coefficient begins at location 149 and ends at location 191. The Op 13 function does *not* mechanize the equation shown on page V-37 of *Personal Programming*, that is, the solution is not obtained by calculating the values for the slope and for the standard deviations of the x inputs and y inputs and then combining as indicated on page V-37 to obtain the answer. Rather, the algebraic equivalent equation is mechanized.

$$R = \frac{\left(\Sigma xy - \frac{\Sigma x \cdot \Sigma y}{N}\right)}{\left[\left(\Sigma x^2 - \frac{(\Sigma x)^2}{N}\right)\left(\Sigma y^2 - \frac{(\Sigma y)^2}{N}\right)\right]^{\frac{1}{2}}}$$

Of course, this will not be the exact computational equivalent due to round-off errors in the calculator. For the problem illustrated on page V-38 of *Personal Programming* the difference is small, only 2.6 E-12.

The Op 14 and Op 15 functions begin at location 000, store the input value in HIR-8, calculate the slope and intercept using the same instructions as those used for the Op 11 function, but branch at the HIR-20 command at locations 045/046. The Op 15 function continues on at location 047 ending at location 057. The Op 14 function continues on from location 058 ending at location 066.

Identification of the Equivalent Constants

As noted earlier the listing for locations 380 through 511 appear to be "garbage" when considered as instructions. But, with the calculator repartitioned to 319.79 (8-2nd-Op-17) then data registers 56 through 71, which cover the same portion of the memory as program locations 384 through 511, will print out as follows with an INV-2nd-List

```
-572.9577951          56
-31.41592654          57
-157.0796327          58
  -9.9999999-71       59
  -9.9999967-70       60
   9.9996667 64       61
   9.9668652 18       62
   7.8539816 44       63
      9.99999 43      64
      0.00001         65
      -9.9995 28      66
   9.9950033-12       67
   9.9999999 99?      68
    9.531018-34?      69
   6.9314718-95?      70
-23.02585093   ?      71
```

Note: You cannot obtain this printout by adding the sequence LRN-INV-2nd-List to the calling sequence defined on page 1. Rather, you must have entered program steps 384 through 511 using the normal LRN mode.

If you recall the values from data registers 69, 70 and 71 you will find that a flashing display does not result. The question marks by the printouts for those data registers from the INV-2nd-List sequence are due to the flashing display which does come from a recall of the value from data register 68.

Bill Skillman is credited with recognizing that these values are constants which would be useful in mechanizing the transcendental functions (see Reference 3). For example, the contents of memory register 57 is $-\pi \times 10$, of memory register 63 is $\pi/4 \times 10$, and of memory register 70 is $\text{Ln}(2) \times 10^{-94}$, etc. In Reference 4 John Van Wye suggested that the constants were not scaled at all, but were fixed point notation with an implied decimal point between the two most significant digits. Using John's suggestion we may obtain the constants by reading each set of eight equivalent instructions in reverse order; for example, for the constant in memory location 56 we read the instructions from 511 backward through 504. Following this convention yields the following table of values, with the equivalent function as identified by Bill Skillman.

Memory Location	Memory Location Contents	Equivalent Function
56	5.729577951308022	18/π
57	3.141592653590012	π
58	1.570796326795022	π/2
59	0.000099999999667	arctan(0.0001)
60	0.000999999666667	arctan(0.001)
61	0.009999666686670	arctan(0.01)
62	0.099668652491200	arctan(0.1)
63	0.785398163397450	π/4 = arctan(l)
64	0.000000999999500	Ln(1.000001)
65	0.000009999950000	Ln(1.00001)
66	0.000099995000333	Ln(1.0001)

Memory Location	Memory Location Contents	Equivalent Function
67	0.000999500333084	Ln(1.001)
68	0.009950330853168	Ln(1.01)
69	0.095310179804325	Ln(1.1)
70	0.693147180559945	Ln(2)
71	2.302585092994012	Ln(10)

All values are correct to 13 places. Some are correct to 16 places. Location 68 ceases to look like an overflow. That apparent overflow when recalling the value as a constant with the INV-2nd-List sequence or with a RCL-68 instruction, and the strange signs and scalings for the other values, seem to be associated with the fact that the three least significant digits carry the sign and exponent information for normal storage of constants.

A Caution on the Use of INV x_m, Op 11, and Op 13

The discussion of the Op 13 function observed that algebraic equivalents are not necessarily computational equivalents in the presence of round-off error. Such use of algebraic equivalents can lead to unexpected results, particularly in the use of the INV x_m, Op 11, and Op 13 functions. For example,. the definition of the standard deviation is the square root of the sum of the squares of the differences between the individual values and the mean value, divided by the number of observations minus one; that is:

$$\sigma_y = \sqrt{\frac{\Sigma(y - \bar{y})^2}{(N - 1)}}$$

In the TI-59 mechanization for the standard deviation the algebraic equivalent is used as indicated on page V-34 of *Personal Programming*.

$$\sigma_y = \sqrt{\frac{\Sigma y^2 - (\Sigma y)^2/N)}{(N - 1)}}$$

This algebraic equivalent allows the calculator to obtain an answer without storing each individual data point, but at the risk of obtaining erroneous results if round-off errors become important. This might happen if the mean is large, but the deviations from the mean are small.

For example, enter the following five values using the Σ+ command: 1.0000067, 1.0000069, 1.0000071, 1.0000073 and 1.0000075. Then 2nd-x_m yields the correct mean of 1.0000071. but INV-2nd- x_m yields a flashing .0000007071 for the standard deviation because round-off has caused the value under the square root sign to be a negative number.

Other Restrictions on Use of Statistics and Conversions Functions

Page V-30 of *Personal Programming* contains the following caution:

> "There are several mathematical sequences that have been programmed into your calculator. These calculations use up to 4 pending operations and 1 level of the subroutine return register. The pending operation registers used by these functions are the same registers used by the alpha-numeric printing operations. Be sure alphanumeric data entered with 2nd-Op-03 or 2nd-Op-04 is printed before using Conversion or Statistics functions."

The availability of the revealed firmware permits finer tuning of the above restrictions. Examination of the firmware shows the following usage of pending operations, hierarchy registers, and parenthesis for the various functions:

Function	*Parenthesis*	*Pending Operations*	*t-Register*	*HIR-7*	*HIR-8*
DMS	4	2			
INV-DMS	2	2			
P→R	1	1	Used	R	θ
INV-P→R	2	2	Used	X	Y
Σ+	0	0	Used	Used	$+x_i*y_i$
INV-Σ+	0	0	Used	Used	$- x_i*y_i$
x_m	1	1	Used		
INV-x_m	2	2	Used		N-1
Op-11	1	2	Used		
Op-12	2	1	Used		Used
Op-13	1	4			
Op-14	2	1	Used		Used
Op-15	2	1	Used		Used

A Caution on the Use of INV-Σ+

The article "Hard-wired Functions" by Don O'Grady in the March/April 1981 issue of *PPX Exchange* (Reference 9) cautions users against difficulty when using INV-Σ+ with the data entered using scientific notation. The problem is a result of the +/- command at location 213, the first instruction encountered in the INV-Σ+ routine. The instruction was intended to change the sign of the entered value. or the sign of the mantissa when in scientific notation. However, if the INV-Σ+ is used immediately after completion of an entry in scientific notation then the +/- changes the sign of the exponent, not the sign of the mantissa.

The problem occurs only with the "live entry" state which 'results from entry of a number in scientific mode. It does not occur with the "live entry" state established by entering the sequence EE-INV-EE with a dead entry state in the display. When in scientific mode that sequence produces a "live entry" state for the mantissa, not for the exponent.

To observe the problem initialize the calculator for statistics entry with the sequence 2nd-Pgm-01-SBR-CLR. Enter 1-EE-3 and press 2nd-Σ+. Enter 1-EE-3 again and press INV-2nd- Σ+. Recall the sum of the two entries with a RCL-01 and see 1.000001 03 in the display, not the zero which was desired.

Special Applications of the Statistics and Conversions Functions

The statistics and conversions functions can sometimes be used to provide other functions with substantial savings in program steps. For example, in one recent application I needed to recall an angle and calculate and store sin θ, $\sin^2 \theta$, cos θ, $\cos^2 \theta$, and sin θ cos θ . My original routine looked something like this:

```
000   76 LBL
001   11  A
002   43 RCL
003   00  00
004   38 SIN
005   42 STO
006   01  01
007   42 STO
008   06  06
009   33 x²
010   42 STO
011   02  02
012   43 RCL
013   00  00
014   39 COS
015   42 STO
016   04  04
017   49 PRD
018   06  06
019   33 x²
020   42 STO
021   05  05
022   92 RTN
```

As written the routine accepts an angle stored in R00 in degrees and places sin θ in R01, $\sin^2 \theta$ in R02, cos θ in R04, $\cos^2 \theta$ in R05 and sin θ cos θ in R06. When I ran into a memory limitation I reprogrammed the routine using the statistics and conversions functions and saved ten program locations. My revised program was:

```
000   76 LBL
001   11  A
002   36 PGM
003   01  01
004   71 SBR
005   25 CLR
006   01  1
007   32 X:T
008   43 RCL
009   00  00
010   37 P/R
011   78 Σ+
012   92 RTN
```

If I had needed one more program location I could have used the sequence Pgm-15-E' instead of Pgm-01-SBR-CLR since I was using the Master Library module.

Clearing Memory Using the Statistics Initialization Routine

In the article "Clearing-Your Memory" in the March/April 1981 issue of *PPX Exchange* Robert Wyer described a method for clearing a limited portion of the memory by using repartitioning.

A more versatile method is available using a portion of the statistics initialization routine. If you use the program sequence XX-Pgm-01-SBR-012 you will clear the memory locations from R01 through RXX. This capability was illustrated on page III-10 of the *TI Programmable 59 Workbook*. Unfortunately, this routine will not work with all of the modules since TI chose to restructure the 01 program for some of the modules.

The **XX-Pgm-01-SBR-012** sequence works with the modules numbered one through seven, that is with the Master, Statistics, Real Estate, Surveying, Navigation, Aviation, Leisure, and Agriculture modules. The sequence **XX-Pgm-01-SBR-004** is required with the modules numbered nine through eleven, that is the Business Decisions, Math Utilities, and Electrical Engineering modules. The sequence **XX-Pgm-01-SBR-007** is required only with Securities Analysis (#8) module. There is no statistics initialization capability with the RPN module (#13) installed. The restructuring of the later modules involved elimination of the diagnostic capability in favor of additional memory for other programs.

The Circular Stepping Phenomenon

If after completing the keyboard sequence described on page 1 for viewing the firmware you press LRN again you can list the firmware on a PC-100. However, you will find that what would become location 488 in the listing as obtained from the SST readout from the display instead reverts to location 039 when listing on the PC-100. The printout is illustrated in the right hand column on page 19. If you let the printer run long enough the printout will repeat that sequence again and again. In *52 Notes* (Reference J) Steve Bepko reported that if the program counter was SST'd past location 488 then additional printer listing could be obtained up through location 503. At what would have become location 504 the circular stepping to location 039 would occur.

Patrick Acosta noted that a code 22 (INV) appeared at locations 488 and 504 (Reference 10). He suggested that the circular listing might result if the codes were really hexadecimal codes (h22). To test that hypothesis I synthesized a code h22 at location 016 of an otherwise clear memory by using the following sequence (again due to Patrick Acosta):

Starting from turn-on, or Cms-CP	Clears memory
GTO-016-LRN-SBR-BST-LRN	Puts code 71 in location 016
10-Op-17-CLR	0 in the display
Pgm-12-SBR-999	Flashing 0. in the display
R/S	Flashing 0. 00 in the display
DMS	Flashing 0 in the display
LRN	016 55 in the display
Ins	016 55 in the display
LRN-RST-CLR	0 in the display
GTO-016-LRN	016 22 in the display
SST	017 02 in the display
SST	018 10 in the display
SST	019 38 in the display
SST	020 30 in the display
SST	021 31 in the display
SST	022 39 in the display
SST	023 71 in the display
SST	024 03 in the display
SST	025 00 in the display
If you check the remainder of the user memory you will find zeroes.	
RST-List	

You should now get the listing shown on the following page. Note the circular stepping each time what would have become location 016 is encountered. If you want more information on the rules for synthesizing hexadecimal codes see Patrick Acosta's writeup on page 15 of volume 6 number 9/10 of *TI PPC Notes*.

```
000   00   0
001   00   0
002   00   0
003   00   0
004   00   0
005   00   0
006   00   0
007   00   0
008   00   0
009   00   0
010   00   0
011   00   0
012   00   0
013   00   0
014   00   0
015   00   0
000   02   2
001   10  E'
002   38  SIN
003   30  TAN
004   31  LRN
005   39  COS
006   71  SBR
007   00   00
008   00   00
009   00   0
010   00   0
011   00   0
012   00   0
013   00   0
014   00   0
015   00   0
000   02   2
001   10  E'
002   38  SIN
003   30  TAN
004   31  LRN
005   39  COS
006   71  SBR
007   00   00
008   00   00
009   00   0
010   00   0
011   00   0
012   00   0
013   00   0
014   00   0
015   00   0
000   02   2
001   10  E'
002   38  SIN
003   30  TAN
004   31  LRN
005   39  COS
```

etc.

Patrick's conjecture that something is different about the code at locations 488 and 504 is supported by the listing in Table VI of Patent No. 4,153,937 (Reference 6). The final character for the constants 0, 13, 14, and 15 in that table all appear as a code C. The equivalent locations in the downloaded firmware on page 18 and 19 are 384, 488, 496, and 504.

Locations 384 and 496 in the downloaded firmware list as code B (key-code 12). In the table in the patent the final character pair for the equivalent locations is 0C, or hexadecimal code h12.

Locations 488 and 504 in the downloaded firmware list as code INV (key-code 22). In the table in the patent the final character pair for the equivalent locations is 1C, or hexadecimal code h22.

The conclusion that it is the h22 code at locations 488 and 504 which cause the circular stepping seems inescapable.

A method for writing the revealed firmware directly onto magnetic cards has not been found to date. To perform calculations with the revealed firmware it is necessary to key in the proper code in the Normal program assembly mode (LRN). However, even then some of the routines will not run due to the existence of the HIR-20 commands which act as Nop in user memory.

Even if it were possible to record the downloaded firmware on magnetic cards the hexadecimal codes at locations 384, 488, 496, and 504 would not survive the transfer to magnetic cards. Hexadecimal codes revert to their normal code equivalent during transfer to the magnetic cards.

The TI-58C Code

Page 9 of the March/April 1981 issue of *PPX Exchange* states.

> "The TI-58/59 code is the same as that for the TI-58C except for steps 000 through 046 (Op12) and steps 250 through 282 (INV-P→R; R→P)..."

Page 20 shows the TI-58C code as listed in *PPX Exchange* and as verified by Patrick Acosta. The differences do not seem important for normal calculations. However, if the differences are not recognized then unexpected results may occur when exercising "quirks" of the calculators. One example is when creating hexadecimal codes as described in V6N6/7p11 and V6N8p3/4 of *TI PPC Notes*. Patrick Acosta reported such a problem in a personal letter.

> "...We'll have to take note of the ROM version in the future when we create hex-codes at any of the affected steps. For instance, Michael Sperber got a Dsz at step 031 if his graphics mode initialization (v6N4/5p6 of *TI PPC Notes*) while I got a RTN. I had thought that this was an error by whoever did the listing."

Comparison of the listings for the TI-58/59 firmware and for the TI-58C firmware shows a CE (code 24) for the 58/59 and a CP (code 29) for the 58C at location 031. This difference of 5 in the firmware code corresponds to the difference of 5 in the code resulting from hexadecimal initialization, that is, a Dsz (code 97) for the 58/59 and' a RTN (code 92) for the 58C.

Other Calling Sequences for Viewing the Firmware

There are many other calling sequences for viewing the firmware. Representative sequences which will work with the Master Library module installed are described in the abstracts for references 1, 7, 8, and 12 on pages 13 through 15. Some of the unusual addresses which were observed led to speculation as to multiple storage for the firmware. However, patent no. 4,153,937 confirms the existence of only 512 unique memory locations.

Calling sequences are also available for other memory modules. One class of such sequences involve the use of unintended code 31's, a technique similar to that used to find fast mode entry techniques with the other memory modules. With the Leisure Library installed you may download the firmware with the sequence.

GTO-004-LRN-Pgm-11-SBR-423-RST-RST-RST-LRN-RST-R/S-LRN

You will see 6546 81 in the display. If you SST you will see the program code for the firmware, but with the addresses elevated by 6546.

References:

1. "Some 58/59 Firmware Revealed", *52 Notes* V3N10P4, October 1978

Steffen Seitz reported the original discovery of unusual keyboard sequences to reveal the 58/59 firmware. His sequence was, with the Master Library module installed and turn-on conditions:

Op9-Pgm-24-R/S-R/S-R/S-99-0P17-GTO-0-R/S-D.MS-LRN

The calculator was then at step 000 of the firmware. Steffen identified the segments of the firmware which mechanized the various statistics and conversions functions, and found that steps 380 through 487 made no sense as instructions. The circular stepping phenomenon was described. The HIR 20 commands at locations 045/046 and at 081/082 were described as RTN commands. Other CROM modules did not provide access to the firmware with the defined sequence..

2. "More on the Revealed Firmware", *52 Notes* V3N11P1/2, November 1978

Dix Fulton and Gordon Wilk reported on additional experimentation with the revealed firmware. Rules for other calling sequences were defined. Speculation that locations 384 through 487 were thirteen registers which provided addresses and labels for the thirteen programs in the firmware was eventually shown to be incorrect. The revealed firmware failed to explain the rounding which occurs at entry to the D.MS or INV-D.MS routines. The operation of HIR 20 as a RTN, a Nop, or a GTO* depending on the function called was properly defined.

3. "Revealed Firmware", *52 Notes* V3N12P5/6, December 1978

Steve Bepko, Maurice Swinnen, Dave Leising, and John Mickelson found ROM code past step 487 to step 575. Steve Bepko found that by SSTing past location 487 to 489 then listing of steps 489 through 503 could be obtained, followed by circular stepping to step 039. Steps 512 through 575 repeated steps 384 through 447. Bill Skillman identified steps 384 through 511 as non-normalized constants used by the transcendental functions.

4. "Revealed ROM Constants", *52 Notes* V4N1P6, January 1979

John Van Wye suggested that the constants used for the transcendental functions are not scaled but that the calculator treats the values as fixed point numbers. Some constants are good to sixteen places. All are good to 13 places.

5. "ROM Constants Update", *52 Notes* V4N3P4, May 1979

John Van Wye provided further discussion of the ROM constants. He found that nine were good to sixteen places, and that the remaining seven were good to from 13 to 15 places. He noted that the constants for ln(10), π, π/2, and 18/π ended in 12 or 22 which might be some sort of code.

References (continued):

6. U.S. Patent 4,153,937 "Microprocessor System Having High Order Capability", May 18, 1979. Assigned to Texas Instruments.

Table VI confirms 'the existence of only 512 unique memory locations in the firmware. Table IV defines the HIR program codes, including HIR 2X as a conditional return. Lines 64 ff on page 14 provide additional discussion of HIR 20.

7. "TI-59 Firmware", TI PPC Notes V5NIP7, January 1980

GESPRO, a German language newsletter, reported additional work with firmware calling sequences by Walter Ulrich. His sequence

OP9-Pgm-24-R/S-R/S-R/S-OP17-GTO-959-OP17-R/S-P/R-LRN

produces 959 55 in the display. If you SST you can advance up through 1374 00. If you try to print the program sequence you will find that it is the firmware, but at apparently different program counter steps. The printer stops printing at step 999 and the display reverts to zero.

8. "Firmware Revisited", TI PPC Notes V5N3P6, April 1980

Additional calling sequences are discussed. One from TISOFT, a Belgian newsletter is:

GTO-004-LRN-Pgm-02-SBR-240-RST-RST-RST-LRN-RST-R/S-LRN

results in a display of 6546 81. If you go out of LRN and list using a printer the printing starts at location 546. Again, you will find that the printout is the firmware previously identified, but at still different program counter steps. The editor, Maurice Swinnen, suggested that the firmware code was repeated nine times for a total of 8000 steps, and provided a formula for defining the address which would appear in the display depending on the calling sequence. (*Note*: This speculation occurred before the existence of U.S. Patent 4,153,937 became known.) The TISOFT sequence appears to make use of the unintended code 31 (LRN) at location 240 of Pgm 02 of the Master Library module. This technique is similar to the use of a similar sequence to provide entry to fast mode--see the article "TI-59 Fast Mode" in the July/August 1981 issue of *PPX Exchange*.

9. "Hard-wired Functions", PPX Exchange V5N2P7, March/April 1981

The article by Don O'Grady was based on a submission by Palmer Hanson. The firmware which mechanizes the statistics and conversions functions is described. The article contains the first reference to the differences between the TI-58/59 firmware, and that for the TI-58C. The article does not discuss the constants used for the transcendental functions or methods to reveal the firmware.

References (continued)

10. Personal Letter from Patrick Acosta, September 16, 1981

Patrick suggested that the circular stepping phenomenon might be associated with hexadecimal h22 codes rather than normal INV (22) codes at locations 488 and 504.

11. "Firmware", *TI PPC Notes* V6N9/10P21, November 1981

Palmer Hanson pointed out that Patent No. 4,153,937 shows that there are only 512 steps of unique code in the firmware, and that table VI of the patent supports Patrick Acosta's idea that there is some hexadecimal code at the end of some of the constants.

12. TISOFT Newsletter, V3N4P7, July/August/September 1981

Yet another method for viewing the firmware is defined. Starting from turn-on:

9-0p-17-STO-00-Pgm-01-A-CLR-DMS-LRN

As with other sequences you ignore the flashing displays. At the end of the sequence you will see 000 82 in the display. You can then SST through the firmware, or press LRN again and list the program.

TITLE TI-59 Software PAGE 16 OF 20

PROGRAMMER _____ DATE _____

TI Programmable **Coding Form**

LOC	CODE	KEY	COMMENTS
000	82	HIR	Stores
001	08	08	Display
002	53	(	Start
003	53	(	for
004	43	RCL	Op 12,
005	06	06	Op 14,
006	75	-	and
007	43	RCL	Op 15
008	04	04	
009	65	×	
010	43	RCL	
011	01	01	
012	55	÷	Finds
013	43	RCL	slope
014	03	03	per
015	54	)	page
016	55	÷	V-37
017	53	(	of
018	43	RCL	Personal
019	05	05	Programming
020	75	-	
021	43	RCL	
022	04	04	
023	33	X^2	
024	55	÷	
025	43	RCL	
026	03	03	
027	54	)	
028	54	)	
029	53	(	
030	53	(	
031	24	CE	
032	65	×	
033	32	X↕T	Uses
034	43	RCL	slope
035	04	04	to
036	94	+/-	find
037	85	+	intercept
038	43	RCL	per
039	01	01	page
040	54	)	V-37
041	55	÷	of
042	43	RCL	Personal
043	03	03	Programming
044	54	)	
045	82	HIR	Branch
046	20	20	
047	53	(	
048	53	(	
049	94	+/-	Op 15
050	85	+	
051	82	HIR	Finds
052	18	18	x_i for
053	54	)	input y_i
054	55	÷	

LOC	CODE	KEY	COMMENTS
055	32	X↕T	
056	54	)	
057	92	RTN	
058	53	(	
059	32	X↕T	
060	65	×	Op 14
061	82	HIR	Finds
062	18	18	y_i for
063	85	+	input x_i
064	32	X↕T	
065	54	)	
066	92	RTN	
067	53	(	Start
068	43	RCL	for
069	04	04	$\bar{x}$ and Op 11
070	55	÷	
071	43	RCL	
072	03	03	Find
073	54	)	$\bar{x}$
074	32	X↕T	and
075	53	(	$\bar{y}$
076	43	RCL	
077	01	01	
078	55	÷	
079	43	RCL	
080	03	03	
081	54	)	
082	82	HIR	Branch
083	20	20	
084	33	X^2	
085	53	(	
086	94	+/-	
087	85	+	Finds
088	43	RCL	∇_y^2
089	02	02	
090	55	÷	
091	43	RCL	
092	03	03	
093	54	)	
094	32	X↕T	Op 11
095	33	X^2	
096	53	(	
097	94	+/-	
098	85	+	
099	43	RCL	Finds
100	05	05	∇_x^2
101	55	÷	
102	43	RCL	
103	03	03	
104	54	)	
105	32	X↕T	
106	92	RTN	
107	53	(	Start
108	53	(	for
109	43	RCL	INV $\bar{x}$

LOC	CODE	KEY	COMMENTS
110	05	05	
111	75	-	
112	43	RCL	
113	04	04	
114	33	X^2	Find
115	55	÷	∇_x
116	43	RCL	and
117	03	03	store
118	54	)	(N-1)
119	55	÷	for
120	53	(	use
121	43	RCL	in
122	03	03	∇_y
123	75	-	
124	01	1	
125	54	)	
126	82	HIR	
127	08	08	
128	54	)	
129	34	fX	
130	32	X↕T	
131	53	(	
132	53	(	
133	43	RCL	
134	02	02	
135	75	-	Find
136	43	RCL	∇_y
137	01	01	
138	33	X^2	
139	55	÷	
140	43	RCL	
141	03	03	
142	54	)	
143	55	÷	
144	82	HIR	
145	18	18	
146	54	)	
147	34	fX	
148	92	RTN	
149	53	(	
150	53	(	
151	43	RCL	
152	06	06	
153	75	-	
154	43	RCL	Op
155	04	04	13
156	65	×	
157	43	RCL	
158	01	01	
159	55	÷	

MERGED CODES

62 [Pgm] [M]	72 [STO] [M]	83 [GTO] [M]
63 [In] [M]	73 [RCL] [M]	84 [Σ] [M]
64 [M] [M]	74 [SUM] [M]	92 [INV] [SBR]

TEXAS INSTRUMENTS INCORPORATED

TI-24151

PROGRAMMER _____ DATE _____

TI Programmable **Coding Form**

LOC	CODE	KEY	COMMENTS
160	43	RCL	
161	03	03	
162	54	)	
163	55	÷	
164	53	(	
165	53	(	
166	43	RCL	
167	05	05	
168	75	-	
169	43	RCL	
170	04	04	Op 13
171	33	X^2	
172	55	÷	
173	43	RCL	
174	03	03	
175	54	)	
176	65	×	
177	53	(	
178	43	RCL	
179	02	02	
180	75	-	
181	43	RCL	
182	01	01	
183	33	X^2	
184	55	÷	
185	43	RCL	
186	03	03	
187	54	)	
188	54	)	
189	34	$\sqrt{X}$	
190	54	)	
191	92	RTN	
192	44	SUM	Start for Σ+
193	01	01	
194	82	HIR	
195	08	08	
196	33	X^2	
197	44	SUM	
198	02	02	
199	32	X↕T	
200	44	SUM	
201	04	04	
202	82	HIR	
203	48	48	
204	82	HIR	
205	07	07	
206	33	X^2	
207	44	SUM	
208	05	05	
209	01	1	
210	61	GTO	
211	02	02	
212	36	36	
213	94	+/-	Start for INV Σ+
214	44	SUM	

LOC	CODE	KEY	COMMENTS
215	01	01	
216	82	HIR	
217	08	08	
218	33	X^2	
219	94	+/-	
220	44	SUM	
221	02	02	
222	32	X↕T	
223	22	INV	
224	44	SUM	
225	04	04	
226	82	HIR	
227	48	48	
228	82	HIR	
229	07	07	
230	33	X^2	
231	94	+/-	
232	44	SUM	
233	05	05	
234	01	1	
235	94	+/-	
236	44	SUM	
237	03	03	
238	82	HIR	
239	37	37	
240	82	HIR	
241	18	18	
242	44	SUM	
243	06	06	
244	82	HIR	
245	17	17	Common End for Σ+ and INV Σ+
246	32	X↕T	
247	43	RCL	
248	03	03	
249	92	RTN	
250	53	(	
251	82	HIR	
252	08	08	
253	33	X^2	
254	85	+	
255	32	X↕T	
256	82	HIR	
257	07	07	
258	33	X^2	
259	54	)	INV P→R
260	34	$\sqrt{X}$	
261	32	X↕T	
262	53	(	
263	53	(	
264	82	HIR	
265	17	17	
266	55	÷	
267	50	I×I	
268	54	)	
269	22	INV	

LOC	CODE	KEY	COMMENTS
270	39	COS	
271	85	+	
272	53	(	
273	82	HIR	INV P→R
274	18	18	
275	55	÷	
276	82	HIR	
277	17	17	
278	54	)	
279	22	INV	
280	30	TAN	
281	54	)	
282	24	CE	
283	92	RTN	
284	53	(	
285	82	HIR	
286	08	08	
287	39	COS	
288	65	×	
289	32	X↕T	
290	82	HIR	
291	07	07	P→R
292	54	)	
293	32	X↕T	
294	53	(	
295	82	HIR	
296	18	18	
297	38	SIN	
298	65	×	
299	82	HIR	
300	17	17	
301	54	)	
302	92	RTN	
303	53	(	
304	53	(	
305	53	(	
306	82	HIR	
307	08	08	
308	59	INT	
309	65	×	
310	06	6	
311	00	0	
312	85	+	
313	53	(	DMS
314	82	HIR	
315	18	18	
316	22	INV	
317	59	INT	
318	65	×	
319	01	1	

MERGED CODES

62 [P] [A]	72 [STO] [A]	83 [GTO] [A]
63 [LRN] [B]	73 [RCL] [B]	84 [B'] [B']
64 [N'] [C']	74 [SUM] [C]	92 [INV] [SBR]

TEXAS INSTRUMENTS INCORPORATED

TITLE TI-59 Software PAGE 18 OF 20

PROGRAMMER _____ DATE _____

TI Programmable **Coding Form**

LOC	CODE	KEY	COMMENTS
320	00	0	
321	00	0	
322	54	)	
323	82	HIR	
324	08	08	
325	59	INT	
326	54	)	DMS
327	65	×	
328	93	.	
329	06	6	
330	85	+	
331	82	HIR	
332	18	18	
333	22	INV	
334	59	INT	
335	54	)	
336	55	÷	
337	03	3	
338	06	6	
339	54	)	
340	92	RTN	
341	53	(	
342	53	(	
343	82	HIR	
344	08	08	
345	59	INT	
346	85	+	
347	82	HIR	
348	18	18	
349	22	INV	
350	59	INT	
351	65	×	
352	93	.	
353	06	6	
354	54	)	
355	65	×	INV
356	01	1	
357	00	0	DMS
358	00	0	
359	54	)	
360	82	HIR	
361	08	08	
362	53	(	
363	53	(	
364	59	INT	
365	85	+	
366	82	HIR	
367	18	18	
368	22	INV	
369	59	INT	
370	65	×	
371	93	.	
372	06	6	
373	54	)	
374	65	×	

LOC	CODE	KEY	COMMENTS
375	93	.	
376	00	0	
377	01	1	
378	54	)	
379	92	RTN	
380	00	0	
381	00	0	
382	00	0	
383	00	0	
384	12	B	
385	40	IND	
386	99	PRT	R71 = Ln(10)
387	92	RTN	
388	50	I×I	
389	58	FIX	
390	02	02	
391	23	LNX	
392	45	Y×	
393	99	PRT	R70 = Ln(2)
394	55	÷	
395	80	GRD	
396	71	SBR	
397	14	D	
398	93	.	
399	06	6	
400	25	CLR	
401	43	RCL	
402	80	80	R69 = Ln(1.1)
403	79	$\bar{x}$	
404	01	1	
405	31	LRN	
406	95	=	
407	00	0	
408	68	NOP	
409	31	LRN	
410	85	+	R68 = Ln 1.01
411	30	TAN	
412	03	3	
413	95	=	
414	09	9	
415	00	0	
416	84	DP*	
417	30	30	
418	33	x^2	R67 = Ln 1.001
419	00	0	
420	95	=	
421	99	PRT	
422	00	0	
423	00	0	
424	33	x^2	
425	03	3	
426	00	0	R66 = Ln 1.0001
427	95	=	
428	99	PRT	
429	09	9	

LOC	CODE	KEY	COMMENTS
430	00	0	
431	00	0	
432	00	0	
433	00	0	
434	95	=	R65 = Ln 1.00001
435	99	PRT	
436	99	PRT	
437	00	0	
438	00	0	
439	00	0	
440	00	0	
441	95	=	
442	99	PRT	R64 = Ln 1.000001
443	99	PRT	
444	09	9	
445	00	0	
446	00	0	
447	00	0	
448	50	I×I	
449	74	SM*	
450	39	39	R63 = arctan(1) = π/4
451	63	EX*	
452	81	81	
453	39	COS	
454	85	+	
455	07	7	
456	00	0	
457	12	B	
458	49	PRD	R62 = arctan 0.1
459	52	52	
460	86	STF	
461	66	66	
462	99	PRT	
463	00	0	
464	70	RAD	
465	66	PAU	
466	68	NOP	R61 = arctan 0.01
467	66	PAU	
468	96	WRT	
469	99	PRT	
470	09	9	
471	00	0	
472	67	EQ	
473	66	PAU	R60 = arctan 0.001
474	66	PAU	
475	99	PRT	
476	99	PRT	
477	99	PRT	
478	00	0	
479	00	0	

MERGED CODES

62	72 STO	83 GTO
63	73 RCL	84
64	74 SUM	92 INV SBR

TEXAS INSTRUMENTS INCORPORATED

TI 24151

PROGRAMMER _____ DATE _____

TI Programmable Coding Form

LOC	CODE	KEY	COMMENTS
480	67	EQ	
481	96	WRT	
482	99	PRT	· R59
483	99	PRT	=
484	99	PRT	arc tan
485	09	9	0.0001
486	00	0	
487	00	0	
488	22	INV	
489	50	I×I	
490	79	☒	R58
491	26	2ND	=
492	63	EX*	π/2
493	79	79	
494	70	RAD	
495	15	E	
496	12	B	
497	00	0	
498	59	INT	R57
499	53	(	=
500	26	2ND	π
501	59	INT	
502	41	SST	
503	31	LRN	
504	22	INV	
505	80	GRD	
506	30	TAN	R56
507	51	BST	=
508	79	☒	18/π
509	57	ENG	
510	29	CP	
511	57	ENG	

LOC	CODE	KEY	COMMENTS

LOC	CODE	KEY	COMMENTS
480	67	EQ	
481	96	WRT	
482	99	PRT	
483	99	PRT	
484	99	PRT	
485	09	9	
486	00	0	
487	00	0	
039	50	I×I	
040	54	)	
041	55	÷	
042	43	RCL	This
043	03	03	illustrates
044	54	)	
045	82	HIR	the
046	20	20	
047	53	(	circular
048	53	(	
049	94	+/-	stepping
050	85	+	
051	82	HIR	anomaly
052	18	18	
053	54	)	
054	55	÷	
055	32	X↕T	
056	54	)	
057	92	RTN	
058	53	(	
059	32	X↕T	
060	65	×	
061	82	HIR	
062	18	18	
063	85	+	
064	32	X↕T	
065	54	)	
066	92	RTN	
067	53	(	
068	43	RCL	
069	04	04	
070	55	÷	
071	43	RCL	
072	03	03	
073	54	)	
074	32	X↕T	
075	53	(	
076	43	RCL	
077	01	01	
078	55	÷	
079	43	RCL	

MERGED CODES

62	72 STO	83 GTO
63	73 RCL	84
64	74 SUM	92 INV SBR

TEXAS INSTRUMENTS INCORPORATED

TI 24151

TITLE TI-59 Software PAGE 20 OF 20

PRGGRAMMER _____ DATE _____

TI Programmable **Coding Form**

LOC	CODE	KEY	COMMENTS
000	82	HIR	
001	08	08	
002	53	(	
003	53	(	
004	43	RCL	
005	06	06	
006	75	-	
007	43	RCL	
008	04	04	
009	65	×	
010	43	RCL	
011	01	01	
012	55	÷	
013	43	RCL	
014	03	03	
015	54	)	
016	55	÷	
017	53	(	
018	43	RCL	
019	05	05	
020	75	-	
021	43	RCL	
022	04	04	
023	33	X^2	
024	55	÷	
025	43	RCL	
026	03	03	
027	54	)	
028	54	)	
029	53	(	
030	53	(	
031	29	CP	
032	65	×	
033	32	X↕T	
034	43	RCL	
035	04	04	
036	94	+/-	
037	85	+	
038	43	RCL	
039	01	01	
040	54	)	
041	55	÷	
042	43	RCL	
043	03	03	
044	54	)	
045	82	HIR	
046	20	20	

LOC	CODE	KEY	COMMENTS

Note: This page shows those parts of the firmware which are unique to the TI 58C.

LOC	CODE	KEY	COMMENTS
250	53	(	
251	53	(	
252	24	CE	
253	55	÷	
254	32	X↕T	
255	82	HIR	
256	08	08	
257	54	)	
258	22	INV	
259	30	TAN	
260	85	+	
261	53	(	
262	82	HIR	
263	18	18	
264	55	÷	
265	50	I×I	
266	54	)	
267	22	INV	
268	39	COS	
269	54	)	
270	24	CE	
271	53	(	
272	32	X↕T	
273	33	X^2	
274	85	+	
275	82	HIR	
276	18	18	
277	33	X^2	
278	54	)	
279	34	$\sqrt{x}$	
280	32	X↕T	
281	92	RTN	
282	00	0	
283	00	0	

MERGED CODES

62	72 STO	83 GTO
63	73 RCL	84
64	74 SUM	92 INV SBR

TEXAS INSTRUMENTS
INCORPORATED

TI 24151